Evaluation of ACE Properties of Traditional SQL and NoSQL Big Data Systems

Maria Teresa Gonzalez-Aparicio
Department of Computing
University of Oviedo
Gijón, Spain
maytega@uniovi.es

Muhammad Younas
School of ECM
Oxford Brookes University
Oxford, United Kingdom
m.younas@brookes.ac.uk

Javier Tuya
Department of Computing
University of Oviedo
Gijón, Spain
tuya@uniovi.es

Rubén Casado
Accenture Digital
Spain
ruben.casado.tejedor@accenture.com

ABSTRACT

Traditional SQL and NoSQL big data systems are the backbone for managing data in cloud, fog and edge computing. This paper develops a new system and adopts the TPC-DS industry standard benchmark in order to evaluate three key properties, availability, consistency and efficiency (ACE) of SQL and NoSQL systems. The contributions of this work are manifold. It evaluates and analyses the tradeoff between the ACE properties. It provides insight into the NoSQL systems and how they can be improved to be sustainable for a more wide range of applications. The evaluation shows that SQL provides stronger consistency, but at the expense of low efficiency and availability. NoSQL provides better efficiency and availability but lacks support for stronger consistency. In order for NoSQL systems to be more sustainable they need to implement transactional schemes that enforce stronger consistency as well as better efficiency and availability.

CCS CONCEPTS

Information systems→ Data management systems → Database management system engines→ Database transaction processing → Data locking

KEYWORDS

Big data; SQL; NoSQL; Riak; TPC-DS, Data consistency.

1. INTRODUCTION

Cloud computing delivers on-demand IT services, such as storage, compute power and servers, over the Internet in order to offer flexibility, scalability and elasticity in service provisioning. Cloud service consumers only pay for the services they use. This reduces their operational and maintenance cost of IT services. The common model of cloud service provisioning is built around data centers where cloud services are centrally stored and managed. In order to alleviate issues of centralized cloud new models of edge and fog computing have been emerged. Edge computing offers users and developers cloud services and resources at the edge of a network or Internet. It delivers compute, storage and data services much closer to end devices and/or end users [1]. Fog computing model can be defined as an additional layer that provides a bridge between edge computing (resources) and the (centralized) cloud. For example, fog computing can help in cloud resource virtualization in order to dynamically distribute workload across different (edge) computing nodes.

Despite the differences between cloud, fog and edge computing models, they all share the need of storing, processing and analysing data for different types of applications. The work presented in this paper focuses on the traditional SQL and NoSQL big data systems which are used by all the three models, cloud, edge and fog computing. It evaluates the three key properties, availability, consistency and efficiency (ACE). Availability means that data is available. For instance, if one node (of a system) is failed or overloaded (with many requests) then data can be accessed from another node. Consistency means that data must remain in consistent state whenever it is updated. Efficiency refers to the process that data is efficiently accessed and/or updated.

Traditional SQL databases have widely been used for a number of years by various organizations and companies. SQL databases (such as MySQL, Oracle) are built using rigorous theoretical and mathematical models such as relational algebra. They follow the principles of data normalization and integrity constraints in order to maintain strong data consistency. SQL database systems have been used for applications that need strong consistency and data integrity constraints, for example, banking applications, customers and products data, online shopping and so on.

NoSQL big data systems (such as Riak, MongoDB, Couch) are relatively new and they do not generally adopt strong theoretical/mathematical models. They give preference to efficiency and availability over data consistency. They do not follow data normalization principles (as in SQL). Instead they follow weaker or eventual consistency model.

NoSQL systems have been used for applications that need high efficiency and availability but weaker consistency. For instance, NoSQL systems are capable of processing hundreds of

SAC'19, April 8-12, 2019, Limassol, Cyprus

https://doi.org/10.1145/3297280.3297474

get the lock first, are able to satisfy their needs and get the required number of units.

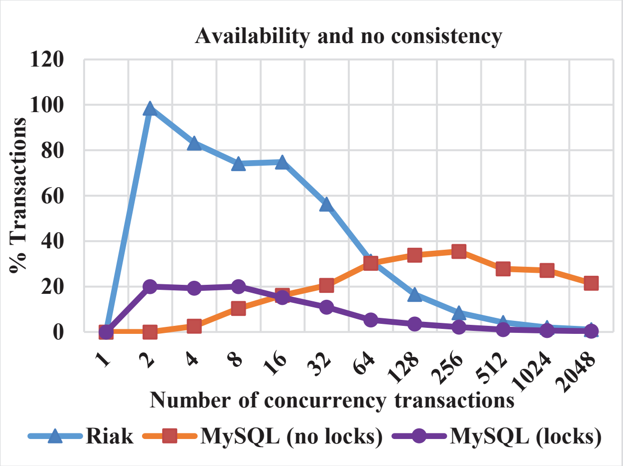

Figure 10. ANC in MySQL and Riak in "Payment" phase.

In MySQL with no locks and in Riak the percentage of transactions that fail due to the shortage of units in the database is very low, because the number of transactions which reach this phase is also low, as it is explained in the Section 4.3.

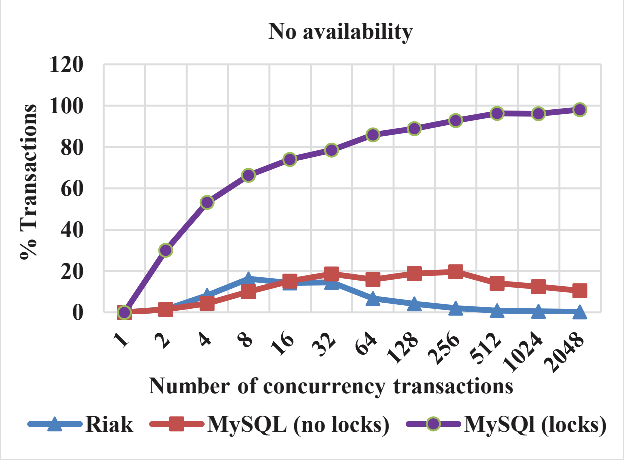

Figure 11. No availability in MySQL & Riak in "Payment" phase.

5. CONCLUSIONS

This paper studied the three main ACE properties, availability, consistency and efficiency (or performance) of the traditional SQL and NoSQL data systems which are used in cloud, fog and edge computing for storing and processing data. In it, we developed a new system using real case study of an online shopping cart and the industry standard benchmark of the TPC-DS in our experiments. We also used the widely used MySQL (traditional database system) and Riak (NoSQL big data system) in the design and implementation of the proposed system and experimentation.

Our work is first that studied the ACE properties of SQL and NoSQL big data systems. It provided greater insights into the strengths and weaknesses of both SQL and NoSQL big data systems. Our extensive experimentation produced various interesting results which show that MySQL with locks provide better consistency. Thus, it is more appropriate for applications that need strong consistency such as online shopping or banking. However, in terms of efficiency and availability Riak outweighs MySQL. But Riak does not ensure strong consistency. Thus, in its current form Riak is not sustainable to be used for applications (such as online shopping or banking). Our recommendation is that Riak needs to support appropriate concurrency control and transaction management mechanisms.

REFERENCES

[1] S. Yi, C. Li, and Q. Li, "A survey of fog computing: concepts, applications and issues," in *Proc. of the 2015 workshop on mobile big data*, 2015, pp. 37-42.

[2] TPC. (2001-2018). Available: http://www.tpc.org/tpcds/

[3] M.T. González-Aparicio, M. Younas, J. Tuya, and R. Casado, "Testing of transactional services in NoSQL key-value databases," *Future Generation Computer Systems,* vol. 80, pp. 384-399, 2018.

[4] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile Networks and Applications,* vol. 19, pp. 171-209, 2014.

[5] M. Cecowski, S. Becker, and S. Lehrig, "Cloud computing applications," in *Engineering Scalable, Elastic, and Cost-Efficient Cloud* Computing *Applications*, ed: Springer, 2017, pp. 47-60.

[6] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers, "Big data: The next frontier for innovation, competition, and productivity," 2011.

[7] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, "The rise of "big data" on cloud computing: Review and open research issues," *Information Systems,* vol. 47, pp. 98-115, 2015.

[8] L. Jiang, M. Jun, and Z. Yang, "Customer-perceived value and loyalty: how do key service quality dimensions matter in the context of B2C e-commerce?," *Service Business,* vol. 10, pp. 301-317, 2016.

[9] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, "Bigtable: A Distributed Storage System for Structured Data," *ACM Trans. Comput. Syst.,* vol. 26, pp. 1-26, 2008.

[10] A. Moniruzzaman and S. A. Hossain, "Nosql database: New era of databases for big data analytics-classification, characteristics and comparison," *arXiv preprint arXiv:1307.0191,* 2013.

[11] A. Lakshman and P. Malik, "Cassandra: a decentralized structured storage system," *ACM SIGOPS Operating Systems Review,* vol. 44, pp. 35-40 2010.

[12] D. G. Campbell, G. Kakivaya, and N. Ellis, "Extreme scale with full SQL language support in microsoft SQL Azure," presented at the Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, Indianapolis, Indiana, USA, 2010.

[13] J. C. Corbett, J. Dean, M. Epstein, A. Fikes, C. Frost, J. J. Furman, S. Ghemawat, A. Gubarev, C. Heiser, P. Hochschild, W. Hsieh, S. Kanthak, E. Kogan, H. Li, A. Lloyd, S. Melnik, D. Mwaura, D. Nagle, S. Quinlan, R. Rao, L. Rolig, Y. Saito, M. Szymaniak, C. Taylor, R. Wang, and D. Woodford, "Spanner: Google's Globally Distributed Database," *ACM Trans. Comput. Syst.,* vol. 31, pp. 1-22, 2013.

[14] W. Lloyd, M. J. Freedman, M. Kaminsky, and D. G. Andersen, "Don't settle for eventual: scalable causal consistency for wide-area storage with COPS," presented at the Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles, Cascais, Portugal, 2011.

[15] J. Baker, C. Bond, J. C. Corbett, J. J. Furman, A. Khorlin, J. Larson, J. M. Leon, Y. Li, A. Lloyd, and V. Yushprakh, "Megastore: Providing Scalable, Highly Available Storage for Interactive Services," in *CIDR*, 2011, pp. 223-234.

[16] K. Chitra and B. Jeevarani, "CLOUD TPS: SCALABLE TRANSACTION IN THE CLOUD COMPUTING," *International Journal of Engineering and Computer Science,* vol. 2, pp. 2280-2285, July, 2013.

[17] R. Jimenez-Peris, M. Patino-Martinez, B. Kemme, I. Brondino, J. O. Pereira, R. Vilaça, F. Cruz, R. Oliveira, and M.Y. Ahmad, "CumuloNimbo: A Cloud Scalable Multi-tier SQL Database," *IEEE Data Eng.Bull.,* vol.38, 2015.

[18] D. Peng and F. Dabek, "Large-scale Incremental Processing Using Distributed Transactions and Notifications," in *OSDI*, 2010, pp. 1-15.

[19] F. Junqueira, B. Reed, and M. Yabandeh, "Lock-free transactional support for large-scale storage systems," in *Dependable Systems and Networks Workshops (DSN-W), 2011 IEEE/IFIP 41st International Conference on*, 2011, pp.176-181.

[20] W. Vogels. (2008). *All things distributed.* Available: https://www.allthingsdistributed.com/2008/12/eventually_consistent.html

[21] C. G. D. J. K. Tim, *Distributed Systems. Concepts and design.*: Addison-Wesley 2001.

[22] H. Berenson, P. Bernstein, J. Gray, J. Melton, E. O'Neil, and P. O'Neil, "A critique of ANSI SQL isolation levels," *SIGMOD Rec.,* vol. 24, pp. 1-10, 1995.